MouseWheelInput PRO



Benutzerhandbuch

Inhaltsverzeichnis

1 WAS IST MWI? 4

2 KOMPATIBILITÄT 4

3 INSTALLATION 4

4 STARTEN 4

5 BENUTZEROBERFLÄCHE 4

6 BEDIENUNG VON MWI 4

6.1 DAS MWI-FENSTER 5
6.2 DAS APPCONTROL-FENSTER 5
6.3 DAS VOLUMECONTROL-FENSTER 6

7 KONFIGURATION 7

7.1 MWI KONFIGURATION 9
7.1.1 UNTERKATEGORIE „MWI SETTINGS" 9
7.1.2 UNTERKATEGORIE „MODE SETTINGS" 10
7.1.3 UNTERKATEGORIE „TIMER SETTINGS" 11
7.1.4 UNTERKATEGORIE „SOUND SETTINGS" 12
7.1.5 UNTERKATEGORIE „DICTIONARY SETTINGS" 12
7.1.6 UNTERKATEGORIE „VOLUME CONTROL SETTINGS" 13
7.1.7 UNTERKATEGORIE „TASKLIST SETTINGS" 13
7.2 APPCONTROL KONFIGURATION 14
7.2.1 UNTERKATEGORIE „CONFIGURATION" 14
7.2.2 UNTERKATEGORIE „APPEARANCE SETTINGS" 15
7.2.3 UNTERKATEGORIE „SÍZE AND LOCATION SETTINGS" 16
7.2.4 UNTERKATEGORIE „CAPTION SETTINGS" 16
7.2.5 UNTERKATEGORIE „EFFECTS SETTINGS" 16
7.2.6 UNTERKATEGORIE „MODE SETTINGS" 17
7.2.7 UNTERKATEGORIE „TIMER SETTINGS" 18
7.3 CCSTICK KONFIGURATION 19
7.3.1 UNTERKATEGORIE „CCSTICK SETTINGS" 19
7.3.2 UNTERKATEGORIE „KEY SETTINGS" 20
7.3.3 UNTERKATEGORIE „BUTTON ASSIGNMENTS" 21
7.4 CAR2PC KONFIGURATION 22
7.4.1 UNTERKATEGORIE „CAR2PC SETTINGS" 23
7.5 POWERMATE KONFIGURATION 24
7.5.1 UNTERKATEGORIE „POWERMATE SETTINGS" 24

8 KONFIGURATION VON AKTIONEN 25

9 KONFIGURATION VON APPLIKATIONEN 27

9.1 BEARBEITEN VON MENÜEINTRÄGEN 29
9.2 NEUE APPLIKATIONEN KONFIGURIEREN 31
9.3 AUTOMATISCHE ÜBERSETZUNG DER TEXTE 32

10 TASTATURBEFEHLE 33

11 PROBLEMBEHANDLUNG 34

11.1 BEI VERWENDUNG VOM MAUSRAD 34
11.2 BEI VERWENDUNG DES POWERMATE 34
11.3 BEI VERWENDUNG DES CCSTICK 34

12 UNTERSTÜTZTE GRAFIK FORMATE 35

13 DOKUMENTHISTORIE 35

14 BILDER VERZEICHNIS 36

15 VERWEISE 36

16 INDEX 37

1 Was ist MWI?

MWI (MouseWheelInput) ist ein Zusatzprogramm welches in erster Linie der Bedienung des Computers über ein externes Medium dient. Als externes Medium kann dabei das Mausrad, der Griffin PowerMate oder der ccStick dienen.
Der ursprüngliche Zweck von MWI war allein die Zeicheneingabe über das Mausrad ohne Verwendung der Tastatur. Seit der PRO Version, in die AppControl eingeflossen ist, können auch andere Applikationen gestartet bzw. kontrolliert werden.
Mittlerweile ist durch das Einbinden des ccStick die Vielfalt der Eingabe enorm angewachsen.
MWI findet mit dem ccStick überall dort Verwendung, wo eine herkömmliche Eingabe mittels Tastatur und Maus nicht möglich bzw. nicht gewünscht ist.

2 Kompatibilität

MWI ist für das Betriebssystem Windows XP entwickelt worden. Andere Betriebssysteme werden offiziell nicht unterstützt, es kann jedoch durch aus sein, dass MWI auf anderen Windows Betriebssystem Versionen funktioniert.
Da MWI unter .NET 1.1 entwickelt wurde, muss das .NET Framework 1.1 ebenfalls auf dem Zielcomputer installiert sein.

3 Installation

Zur Installation von MWI auf einem neuen Computer reicht *es MWIProSetup.zip* zu extrahieren und *Setup.msi* auszuführen.

4 Starten

Nach der Installation wird eine neue Programmgruppe im Startmenü angelegt. Weiterhin wird eine Verknüpfung in den Autostart-Ordner eingetragen. Somit wird nach jedem Neustart von Windows MWI automatisch ausgeführt.
Ist MWI gestartet, erscheint ein Symbol ( ) in der Taskleiste.

5 Benutzeroberfläche

Die Standard Sprache von MWI ist English. Eine Umschaltung zu einer anderen Sprache ist nicht möglich. Allerdings sind die Texte einfach gehalten und leicht verständlich.

6 Bedienung von MWI

Zur Bedienung von MWI gibt es ein paar Flash Filme, die den Umgang live zeigen. Die Filme können hier abgerufen werden:
http://zhorvat.dyndns.org/PCForums/ShowPost.aspx?PostID=2055

6.1 Das MWI-Fenster

Das MWI-Fenster wird durch langen Klick des Mausrades bzw. Der Menütaste zum Vorschein gebracht und dient der Eingabe von Zeichen.
Der Aufbau dieses Fensters ist relativ simpel:



Abbildung 1: MWI-Fenster

Durch Drehen des Rades wird die jeweils nächste / vorhergehende Zeichen ausgewählt.
Durch einen Klick des Rades wird das Zeichen an die Applikation im Vordergrund gesendet.
Sofern die Verwendung eines Wörterbuches konfiguriert ist, wird ab dem 2. Buchstaben die Liste der verfügbaren Begriffe in einem separaten Fenster gezeigt. Mit jedem weiteren Buchstaben wird die Liste eingeschränkt.
Um einen Begriff aus der Liste auszuwählen, muss gewartet werden bis das Wörterbuch den Fokus bekommen hat - das passiert automatisch- dann kann der passende Begriff ausgewählt werden und per Klick des Rades übernommen werden.

Ist die Sprachausgabe eingeschaltet, wird der aktuell selektierte Buchstabe bzw. Begriff im Wörterbuch vorgelesen.

6.2 Das AppControl-Fenster

Das AppControl-Fenster wird druch einen kurzen Klick des Mausrades bzw. der Menütaste zum Vorschein gebracht und dient der Steuerung von Applikationen.

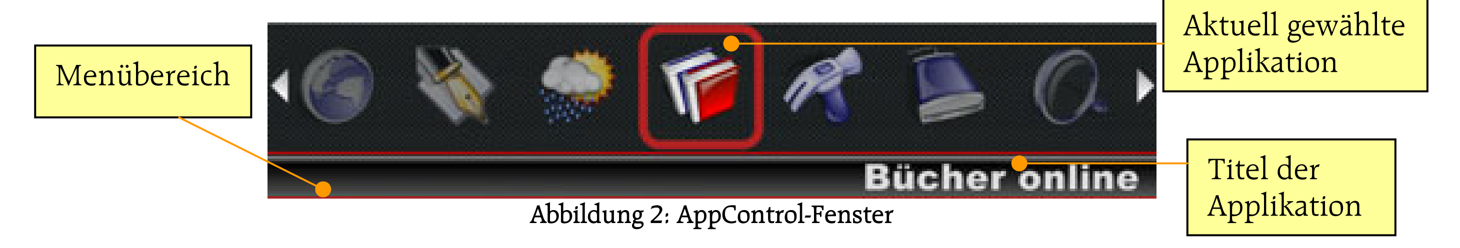


Abbildung 2: AppControl-Fenster

Durch Drehen des Rades wird die jeweils nächste / vorhergehende Applikation ausgewählt.
Durch einen Klick des Rades wird das Menü der Applikation angezeigt oder die (einzige definierte) Aktion ausgeführt.

Nach der Ausführung der Aktion schließt das Fenster sich selbsttätig.

6.3 Das VolumeControl-Fenster

Das VolumeControl-Fenster wird durch einen Doppelklick des Mausrades bzw. der Menütaste zum Vorschein gebracht.
Es dient einzig der Regelung der Lautstärke der Konfigurierten Quelle.

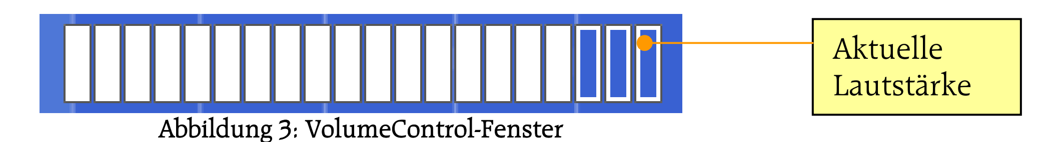


Abbildung 3: VolumeControl-Fenster

Durch Drehen des Rades wird die Lautstärke verringert bzw. erhöht.

7 Konfiguration

Damit MWI nach den Vorstellungen des jeweiligen Benutzers funktioniert, muss es zunächst nach dem ersten Start konfiguriert werden. Dazu reicht ein Klick mit der rechten Maustaste auf dieses Symbol. Es erscheint ein Menü, in dem unter anderen ein Menüpunkt *Configuration* auftaucht. Ein Klick auf diesen Menüpunkt startet die Konfiguration. Es erscheint folgendes Fenster:

Sortierung der Eigenschaften

Kategorie

Name der Unterkatgorie

Tabelle der Eigenschaften

Beschreibung der aktuell gewählten Eigenschaft

Gruppierung von Eigen-schaften

Layout Vorschau

Schriftgröße umschalten

Änderungen bestätigen bzw. widerrufen

MouseWheelInput Configuration

MWI | AppControl | ccStick | Car2PC | PowerMate | can2com

a. MWI Settings

BackgroundColor	Black
BackgroundImage	C:\Data\MouseWheelInput\Background
DynamicPosition	False
Font	Arial Black; 58pt
ForegroundColor	Red
InvertSelectedChar	True
Letters	ABCDEFGHIJKLMNOPQRSTUVWXYZ 12345678
Location	0; 0
Opacity	100
SelectionLineWidth	4
SelectionRectRadius	30
Size	800; 100
UseAnimation	True

b. Mode Settings

BackgroundColor
The background color for the MWI (and dictionary) window (will be overridden by property BackgroungImage).

ESC A B C D E F

Switch font size | OK | Cancel

Abbildung 4: MWI Konfigurations-Fenster

Das Konfigurationsfenster hat immer die gleiche Ansicht, egal welche Kategorie gerade konfiguriert wird (abgesehen vom Layout-Vorschau Bereich; dieser ist nur unter MWI und AppControl vorhanden).

Basis ist immer die tabellarische Darstellung der Eigenschaften im Format *Name der Eigenschaft*, *aktuelle gesetzter Wert der Eigenschaft*. Einzelne Eigenschaften sind in Unterkategorien zusammengefasst, die eine logische Gruppierung vornehmen.
Die genaue Beschreibung der aktuell ausgewählten Eigenschaft findet unterhalb der Tabelle statt. Hier werden auch zusätzliche Informationen als Hilfestellung zum setzen der Eigenschaft angezeigt.
Eigenschaften mit einem vorangestellten Plus Zeichen (+) kennzeichnen eine Gruppierung. Durch einen Klick auf das Plus kann eine Gruppierung expandiert werden.

Eigenschaften werden verändert, indem der neue Wert in der zweiten Spalte eingegeben wird. Hier kann es sein, dass ein weiterführender Dialog zum setzen der (komplexen) Eigenschaft geöffnet und angezeigt wird.
Bei Eigenschaften die das Layout betreffen wird die Änderung automatisch in der Layout-Vorschau wiedergegeben.

Nachdem die Einstellungen vorgenommen wurden, müssen sie mit *OK* bestätigt werden oder mit *Cancel* widerrufen werden (die zuvor eingestellten Parameter bleiben gültig).



7.1 MWI Konfiguration

7.1.1 Unterkategorie „MWI Settings“

In dieser Kategorie finden sich Einstellungen zum Erscheinungsbild des MWI Fenster.

BackgroundColor

Definiert die Farbe für den Fensterhintergrund. Diese ist nicht mehr sichtbar, wenn die Eigenschaft BackgroundImage gesetzt wird, allerdings dient sie als Vordergrundfarbe wenn InvertSelectedChar gesetzt ist.

BackgroundImage

Definiert die die Grafik die im Fensterhintergrund gezeichnet werden soll. Die Grafikdatei kann unterschiedliche Formate haben (JPG, GIF, TIFF, PNG), allerdings wird empfohlen PNG Dateien zu verwenden,
Die Grafik wird im Gegensatz zu der in AppControl nicht an die Größe des Fensters angepasst.

DynamicPosition

Weist an, das MWI Fenster relativ zur aktuellen Position des Mauszeigers zu platzieren. Die ist jedoch nur dann Sinnvoll, wenn das MWI Fenster nicht auf die aktuelle Desktopbreite eingestellt ist.

Font

Definiert den Zeichensatz der zum darstellen der wählbaren Zeichen dienen soll. Hierbei ist zu beachten, dass lediglich **Name** und **Size** eingestellt werden können. Alle anderen Font-Eigenschaften werden von MWI nicht berücksichtigt-

ForegroundColor

Beschreibt die Farbe in der die Zeichen dargestellt werden.

InvertSelectedChar

Weist an, das aktuell ausgewählte Zeichen (das in der Mitte) invertiert darzustellen um eine Hervorhebung zu bewirken.

Letters

Beschreibt welche Zeichen zur Auswahl stehen sollen. Es ist keine Limitierung der Anzahl der Zeichen vorgegeben. Die hier definierten Zeichen werden als „Endlosband" dargestellt. Es ist erlaubt, das häufige Zeichen (wie das Leerzeichen) mehrfach vorkommen.
Eine besondere Bedeutung haben die Zeichenkombinationen „\n", „\t", „\b" und „\x": Sie definierten die Sonderzeichen *Eingabe*, *Tabulator*, *Rückschritt* und *Escape*. Diese Zeichen werden im MWI Fenster gesondert dargestellt als: ↵ ⇥ ← ESC

Location

Definiert an welcher Position das MWI-Fenster erscheinen soll. Diese Eigenschaft hat keine Bedeutung wenn *DynamicPosition* auf **true** gesetzt ist.



Opacity

Beschreibt den Grad der Undurchsichtigkeit des Fensters. Dabei ist 100 komplett undurchsichtig und 0 komplett durchsichtig. Wird dieser Wert auf beispielsweise 75 gesetzt, scheint der Hintergrund leicht hindurch. Dies kann in einigen Fällen nützlich sein (z.B. wenn *DynamicPosition* gesetzt ist).

SelectionLineWith

Definiert welche Breite der Rahmen um das aktuell ausgewählte Zeichen haben soll. Diese Einstellung ist bedeutungslos, wenn *InvertSelectedChar* auf true gesetzt ist.

SelectionRectRadius

Beschreibt welchen Radius die gerundeten Ecken des Rahmens um das aktuell ausgewählte Zeichen haben sollen. Wird dieser Wert auf 0 gesetzt, erscheint ein eckiger Rahmen.

Size

Definiert die Größe des MWI-Fensters.

UseAnimation

Beschreibt ob das Fenster beim erscheinen animiert werden soll. Für langsame Grafikkarten empfiehlt es sich, diese Eigenschaft auf false zu setzen.

7.1.2 Unterkategorie „Mode Settings“

In dieser Unterkategorie wird festgelegt, welche Optionen benutzt werden sollen. Hierbei ist eine Kombination durchaus möglich, wenngleich nicht immer Sinnvoll.

UseAppControl

Definiert ob AppControl verwendet werden soll. Wenn dieser Wert auf false gesetzt ist, erscheint bei einfachem Klick das AppControl Fenster nicht. Unter Verwendung des Mausrades steht somit der Mausrad-Klick weiterhin anderen Applikationen zur Verfügung.

UseCan2Com

Definiert ob das Can2Com Interface benutzt werden soll. Diese Option ist für Künftige Erweiterungen geplant.

UseCar2PC

Definiert, ob das Car2PC Interface benutzt wird. Diese Implementierung hat nichts mit der Steuerung von MWI zu tun, sie dient lediglich der Erweiterung der Eingabemöglichkeiten.
Detaillierte Informationen zu diesem Interface gibt es unter http://www.IndashPC.org

UseCcStick

Definiert ob das ccStick Interface zur Steuerung von MWI verwendet werden soll. Detaillierte Informationen zum ccStick gibt es hier:
http://zhorvat.dyndns.org/PCForums/ShowForum.aspx?ForumID=72

UseMouseHook

Definiert ob das Mausrad zur Eingabe verwendet werden soll.

UseOnClick

Beschreibt, ob bei einem langen Klick die **OnClick** Aktion in (Kapitel 7.2.6) von AppControl ausgeführt werden soll anstatt das MWI-Fenster anzuzeigen.

UsePowerMate

Definiert ob der Griffin PowerMate zur Steuerung von MWI verwendet werden soll.

7.1.3 Unterkategorie „Timer Settings“

In dieser Kategorie werden die Timer (Zeitgeber) Einstellungen vorgenommen.

DoubleclickTime

Definiert innerhalb welcher Zeitspanne zwei Mal geklickt werden muss um einen Doppelklick zu signalisieren. Der Doppelklick dient z.B. dem Anzeigen des Lautstärke Regler Fensters.

InvokeTime

Beschreibt wie lange das Rad (bei ccStick die Menu-Taste) gehalten werden muss um einen „langen Klick" zu signalisieren. Dieser Wert wird in Millisekunden angegeben.

RevokeTime

Beschreibt nach welcher Zeitspanne sich das MWI-Fenster automatisch wieder schließt. Dieser Wert wird in Millisekunden angegeben.

SlowDown

Dieser Wert dient zum "abbremsen" des PowerMate. Je höher der hier eingestellte Wert, desto weniger empfindlich reagiert MWI auf die Signale vom PowerMate.

SwitchToDictionaryTime

Dieser Wert legt fest, nach welcher Zeitspanne (in Millisekunden) der Fokus vom MWI Fenster an das Dictionary Fenster übergeben bzw. zurückgegeben wird.
Diese Zeitspanne muss unter der ***RevokeTime*** liegen, da sonst das MWI-Fenster geschlossen wird, bevor der Fokus an das Wörterbuch gegeben werden kann.

VoiceTime

Dieser Wert beschreibt nach welcher Ruhezeit (keine Eingaben finden über Mausrad, PowerMate, ccStick statt) in Millisekunden die Sprachausgabe aktiviert wird. Dieser Wert ist nur relevant, wenn auch *UseVoice* auf **true** gesetzt ist.

7.1.4 Unterkategorie „Sound Settings“

EnableSound

Weist an, dass die Soundeffekte verwendet werden sollen. Diese Option sollte ausgeschaltet werden, wenn die Grafiken von MWI ins stocken geraten.

UseVoice

Definiert ob die Sprachausgabe aktiviert werden soll. Es wird die unter Windows XP vorgegebene Sprachausgabe / Stimme verwendet. Da Windows XP nur Englische Stimmen mitbringt, empfiehlt es sich andere Stimmen zu installieren, wenn diese Option aktiviert wird,

7.1.5 Unterkategorie „Dictionary Settings“

Dictionary

Beschreibt welche Datei als Wörterbuch verwendet werden soll. Ist diese Eigenschaft leer, so wird vom Wörterbuch kein Gebrauch gemacht.
Das Wörterbuch hat keine eigene Einstellung für die Schrift. Es werden die Einstellungen vom MWI-Fenster übernommen.

DictionaryAppendSpace

Definiert, ob nach dem auswählen eines Begriffes aus dem Wörterbuch ein Leerzeichen angehängt werden soll oder nicht.

DictionaryBackgroundColor

Definiert die Hintergrundfarbe des Fensters.

DictionaryBackgroundImage

Definiert das Hintergrundbild des Fensters (überschreibt *DictionaryBackgroundColor*).

DictionaryForegroundColor

Definiert die Vordergrundfarbe des Fensters (Textfarbe).

DictionaryOpacity

Siehe Beschreibung für *Opacity* in 7.1.1.

DictionaryPosition

Definiert die Position für das Wörterbuch, die entweder rechts daneben (right) oder unterhalb (below) sein kann.

MaxDictionaryList

Definiert wie viele Einträge maximal in dem Wörterbuch zu sehen sein sollen.



7.1.6 Unterkategorie „Volume Control Settings“

ControlVolume

Definiert welcher Audiokanal kontrolliert werden soll. Steht die Einstellung auf None, so wird das Fenster nie angezeigt.

VolumeColor

Definiert welche Farbe für das Zeichnen der Lautstärke Regelung verwendet wird.

VolumeOpacity

Siehe Beschreibung für *Opacity* in 7.1.1.

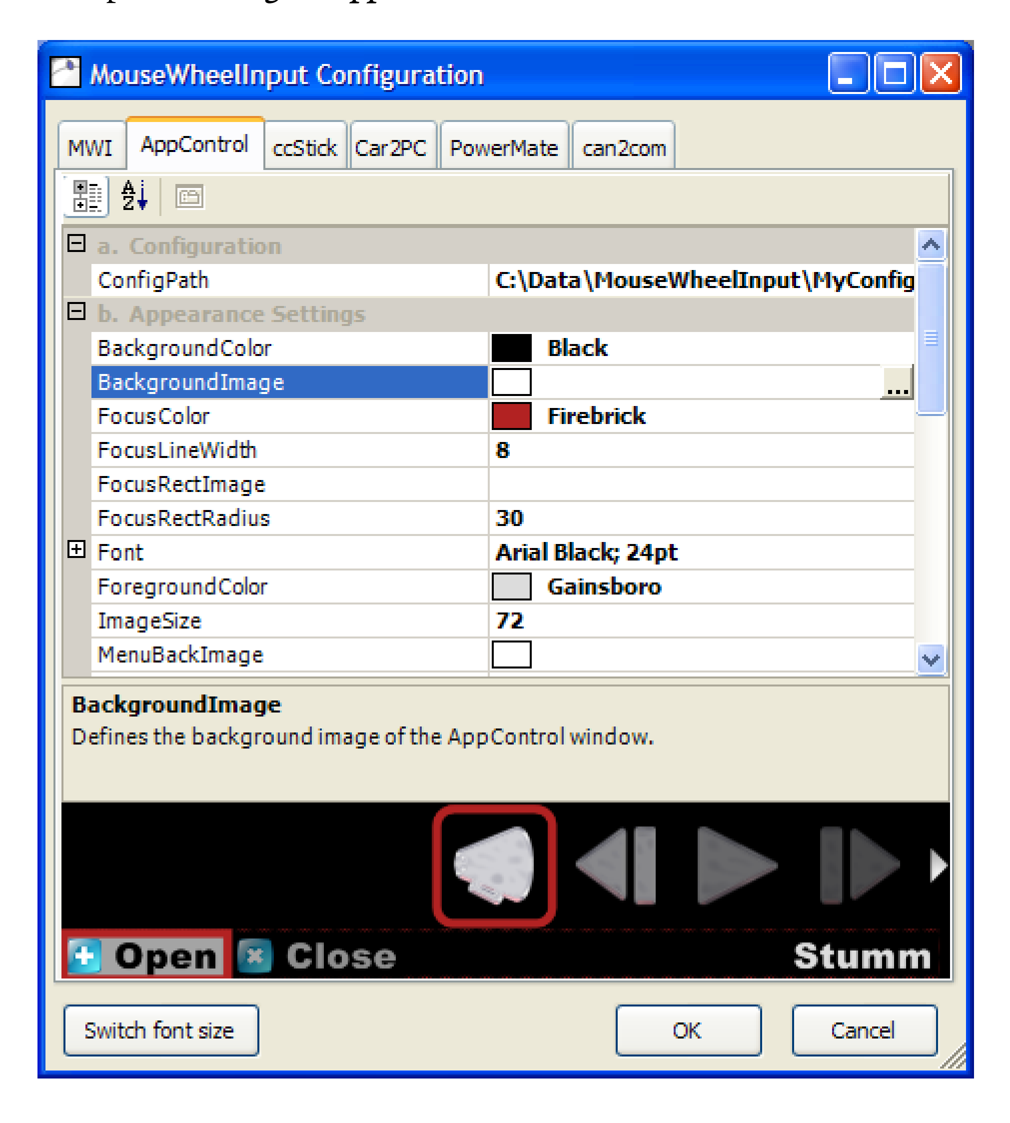
7.1.7 Unterkategorie „Tasklist Settings“

ExcludeFromTaskList

Definiert eine Liste von Anwendungen, die nicht in der Taskliste gelistet werden sollen (gilt in erster Linie für Hintergrund Applikationen).
Für die Eingabe wird ein gesonderter Dialog verwendet, in dem die Namen der Prozesse, so wie sie im Task-Manager (unter Prozesse) angezeigt werden, eingegeben werden. Jeweils ein Prozessname pro Zeile.

7.2 AppControl Konfiguration

AppControl ist ein weiterer Bestandteil von MWI und dient dazu Applikationen zu starten, in den Vordergrund zu holen bzw. mit gezielten Befehlen zu steuern.
Eine Möglichkeit der Steuerung ist beispielsweise das senden von Tastaturkommandos an die jeweilige Applikation.
Welche Applikationen im Fenster aufgelistet werden ist frei konfigurierbar (die Konfiguration kann auf einen Doppelklick im der Layout-Vorschau gestartet werden, oder über den Menüpunkt *Configure Applications*).



7.2.1 Unterkategorie „Configuration“

ConfigPath

Diese Eigenschaft definiert aus welchem Pfad die Konfigurationen für die Applikationen gelesen werden. Die Gold-Edition von MWI ist in der Lage, per Kommando zwischen mehreren Konfigurationen umzuschalten.

Während der Installation ist im Installationsverzeichnis (*C:\Programme\Zoran Horvat\MouseWheelInput*) ein Unterverzeichnis mit dem Namen **MyConfigs** angelegt worden, welches einige nützliche Konfigurationen bereithält.
Es empfiehlt sich eigene Konfigurationen bzw. Änderungen an der Standard Konfiguration in einem eigenen Verzeichnis vorzunehmen.

7.2.2 Unterkategorie „Appearance Settings"

In dieser Kategorie werden die Eigenschaften für das Aussehen des AppControl-Fensters definiert.

BackgroundColor

Definiert die Hintergrundfarbe des Fensters. Diese Eigenschaft wird von BackgroundImage überschrieben, dient dann aber immer noch dem anzeigen des aktuellen Menüpunktes.

BackgroundImage

Definiert die Bild das in den Fensterhintergrund gezeichnet werden soll. Im Gegenteil zu derselben Eigenschaft im MWI-Fenster, wird die Grafik skaliert, also an die Größe des Fensters angepasst.

FocusColor

Bestimmt welche Farbe der Fokus Rahmen der aktuell ausgewählten Applikation haben soll. Diese Eigenschaft wird durch **FocusRectImage** überschrieben, dient jedoch dann noch dem anzeigen des aktiven Menüpunktes.

FocusLineWidth

Definiert die Linienbreite des Fokus Rahmens.

FocusRectImage

Definiert welches Bild für die aktuell selektierte Applikation verwendet werden soll.

FocusRectRadius

Definiert welchen Radius der Fokus Rahmen haben soll. Wird hier eine 0 eingetragen, erscheint der Rahmen rechteckig.

Font

Definiert den Zeichensatz der zum darstellen der Texte dienen soll. Hierbei ist zu beachten, dass lediglich **Name** und **Size** eingestellt werden können. Alle anderen Font-Eigenschaften werden nicht berücksichtigt.

ForegroundColor

Definiert die Vordergrundfarbe (Textfarbe) die zum zeichnen verwendet werden soll.

ImageSize

Spezifiziert welche Größe die Symbole der Applikation (Icons) haben sollen. Die Größe ist frei wählbar und die Symbole werden entsprechend skaliert. Zu beachten ist hierbei, dass das Fenster ebenfalls in der Größe angepasst werden muss.



MenuBackImage
Definiert ein Hintergrundbild, welches für die Menüpunkte einer Applikation gezeichnet werden soll. Diese Einstellung überschreibt den spezifischen Menühintergrund der zu einer Applikation definiert wurde.

Opacity
Siehe Beschreibung für *Opacity* in 7.1.1.

7.2.3 Unterkategorie „Síze and Location Settings“

Location
Definiert an welcher Position das AppControl-Fenster erscheinen soll.

Size
Definiert die Größe des Fensters.

7.2.4 Unterkategorie „Caption Settings“
Diese Kategorie definiert die Eigenschaften der Titelleiste innerhalb des AppControl-Fensters.

CaptionAlignment
Definiert wie die Titelleiste innerhalb der angegebenen Größe (**Size**) angeordnet werden soll.

CaptionLocation
Definiert an welchem Punkt die Titelleiste erscheinen soll. Da die Berechnung der Symbolposition von dieser Position abhängt, muss die Titelleiste immer unterhalb des Symbols der Applikation positioniert sein. Ein gestrichelter Rahmen zeigt in der Layout-Vorschau sowohl Lage als auch Dimension der Titelleiste an.

CaptionSize
Definiert die Größe der Titelleiste. Ein gestrichelter Rahmen zeigt in der Layout-Vorschau sowohl Lage als auch Dimension der Titelleiste an.

7.2.5 Unterkategorie „Effects Settings“
In dieser Kategorie sind die Eigenschaften für die Effekte.

ShowArrows
Definiert ob die Pfeile als Indikatoren für weitere Applikationen angezeigt werden sollen. Dies ist relevant, da AppControl Applikationen nicht als Endlosband angezeigt werden wie die Zeichen im MWI–Fenster.

UseAnimation
Definiert ob das Fenster animiert erscheinen / verschwinden soll (schön für's Auge).



UseFading

Definiert ob die Symbole vor bzw. hinter dem der aktuell selektierten Applikation verblassen sollen.

UseHorizontalMenu

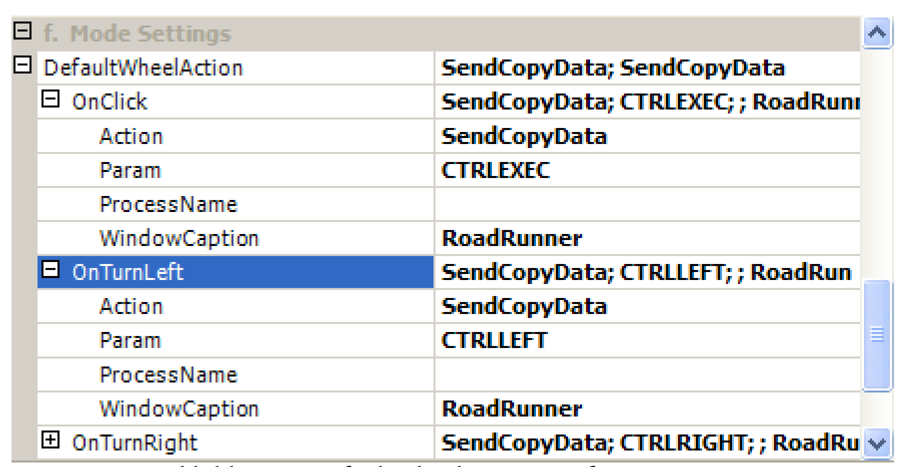
Definiert ob eine horizontales Menü oder ein vertikales Menü verwendet werden soll. Das Horizontale bietet den Vorteil, dass es nicht durch den unteren Bildschirmrand eingeschränkt ist.

7.2.6 Unterkategorie „Mode Settings“

Hier befinden sich Eigenschaften die sich auf den Betriebsmodus auswirken.

DefaultWheelAction

Diese gruppierte Eigenschaft beschreibt was passiert, wenn das Mausrad / PowerMate / ccStick im Ruhemodus (also bei nicht aktiviertem MWI-, AppControl- oder VolumeControl-Fenster) betätigt wird.
Auch hierfür können spezifische Aktionen hinterlegt werden, die dazu dienen andere Applikationen zu kontrollieren. Dazu sind die Eigenschaften OnClick, OnTurnLeft und OnTurnRight vorgegeben:



f. Mode Settings	
DefaultWheelAction	SendCopyData; SendCopyData
OnClick	SendCopyData; CTRLEXEC; ; RoadRunı
Action	SendCopyData
Param	CTRLEXEC
ProcessName	
WindowCaption	RoadRunner
OnTurnLeft	SendCopyData; CTRLLEFT; ; RoadRun
Action	SendCopyData
Param	CTRLLEFT
ProcessName	
WindowCaption	RoadRunner
OnTurnRight	SendCopyData; CTRLRIGHT; ; RoadRu

Abbildung 5: DefaultWheelAction Konfiguration

Wie in der oberen Abbildung zu sehen ist, müssen für OnClick, OnTurnLeft und OnTurnRight die Eigenschaften *Action*, *Param*, *ProcessName* und *WindowCaption* zugewiesen werden. Weitere Details hierzu gibt es im Kapitel 8.

EnableScripting

Definiert ob die Verarbeitung von Javascript / VB-Script Skripts zulässig ist.

SwapWheelDirection

Definiert ob die Drehrichtung vertauscht werden soll. Diese Eigenschaft ist wiederum gruppiert und hält die Eigenschaften *InBar* und *InMenu*.

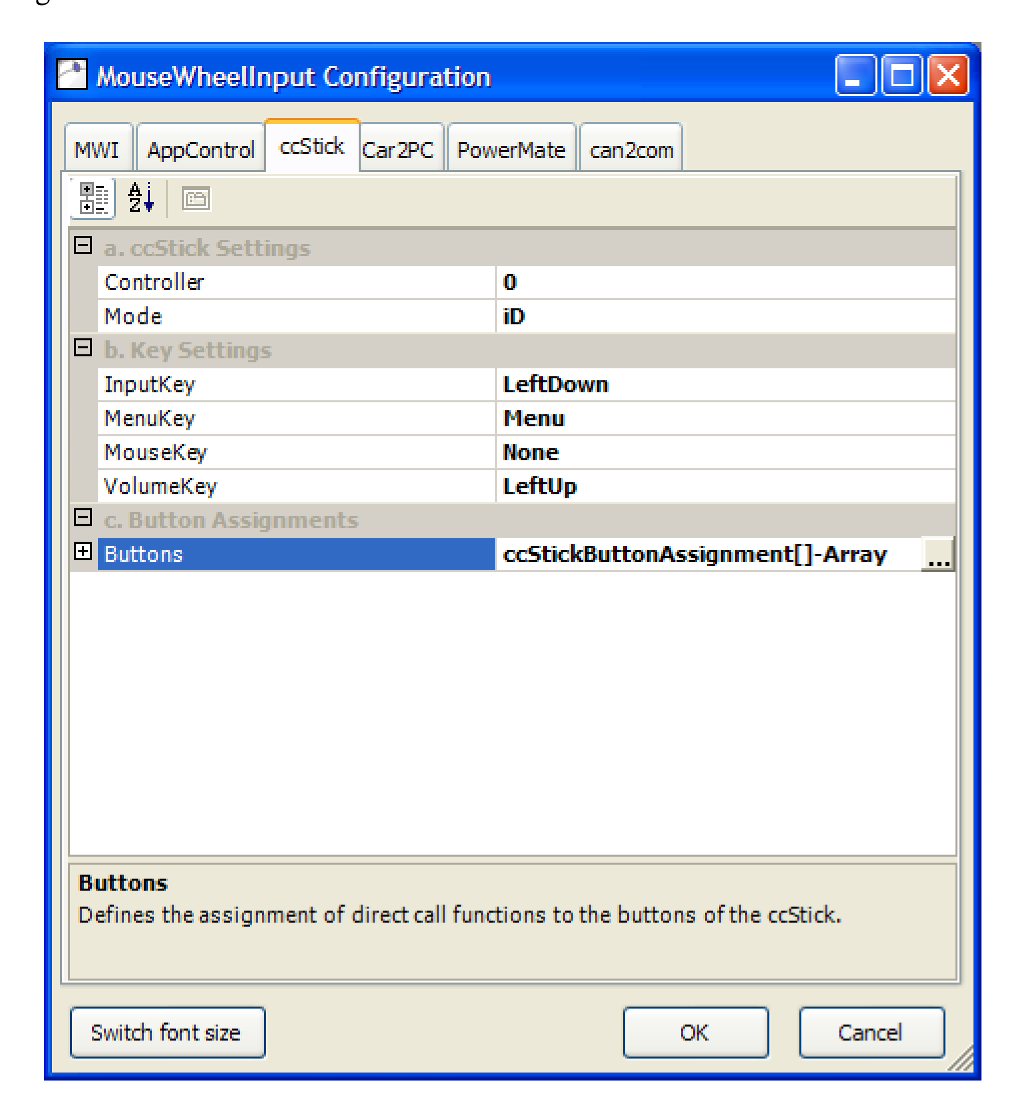
7.2.7 Unterkategorie „Timer Settings“

HideTime

Definiert nach welcher Zeitspanne in Millisekunden das AppControl-Fenster automatisch wieder verschwindet.

7.3 ccStick Konfiguration

In dieser Kategorie sind die Einstellungen für den ccStick zu finden.
Wichtig! Diese Einstellungen werden nur berücksichtigt wenn eine **Gold** Lizenz vorliegt und ein ccStick angeschlossen ist. Anderenfalls sind die Einstellungen hier ohne Bedeutung für MWI.



7.3.1 Unterkategorie „ccStick Settings“

Controller

Definiert welcher Controller als Eingabemedium ausgewertet werden soll. Die ist dann relevant, wenn mehrere ccStick Platinen am Rechner angeschlossen sind. Wird dieser Wert auf 0 gesetzt, so wird der zuerst angesteckte Controller verwendet.

Mode

Definiert welche Variante des ccStick eingesetzt wird. Hier gibt es nur die Einstellung **Standard**, **iD** oder **Mini**. Diese Einstellung muss mit der tatsächlich verwendeten

Hardware übereinstimmen, sonst funktioniert die Auswertung der Signale nicht richtig.

7.3.2 Unterkategorie „Key Settings“

Das der ccStick noch zusätzliche Tasten haben kann, können hier Zuweisungen vorgenommen werden, um diverse Funktionen direkt anzusteuern. Werden Tasten hier belegt, haben sie in Buttons keine Auswirkung mehr.

InputKey

Definiert welche Taste dazu verwendet werden soll das MWI-Fenster direkt aufzurufen.

MenuKey

Definiert welche Taste dazu verwendet werden soll das MWI-Fenster bzw. AppControl-Fenster aufzurufen (einfacher Klick / langer Klick). Diese Belegung sollte immer auf Menu stehen, kann aber im Bedarfsfall verändert werden.
Die Menu Taste ist beim ccStick beschriftet, beim ccStick iD ist dies der Taster der mit Pin 1 des IOWarrior Chips verbunden ist.

MouseKey

Definiert welche Taste dazu verwendet werden soll um den Maus-Modus ein bzw. aus zu schalten. Diese Umschaltung ist nur für den ccStick iD relevant, da bei diesem der Joystick als Kreuzcursor verwendet wird. Ist der Maus-Modus aktiv, wird mit drücken des Rades ein linker Mausklick ausgelöst. Nach 20 Sekunden Ruhemodus schaltet der Maus-Modus sich automatisch wieder ab.

VolumeKey

Definiert welche Taste dazu verwendet werden soll das VolumeControl-Fenster direkt aufzurufen.

7.3.3 Unterkategorie „Button Assignments"

In dieser Unterkategorie ist die Zuordnung der Tasten zu spezifischen Funktionen.

Buttons

Diese Eigenschaft ist eine Sammlung an Tasten-Aktionen. Ist die Gruppierung aufgeklappt, werden alle Definitionen aufgelistet. Zum Bearbeiten erscheint ein separater Dialog:

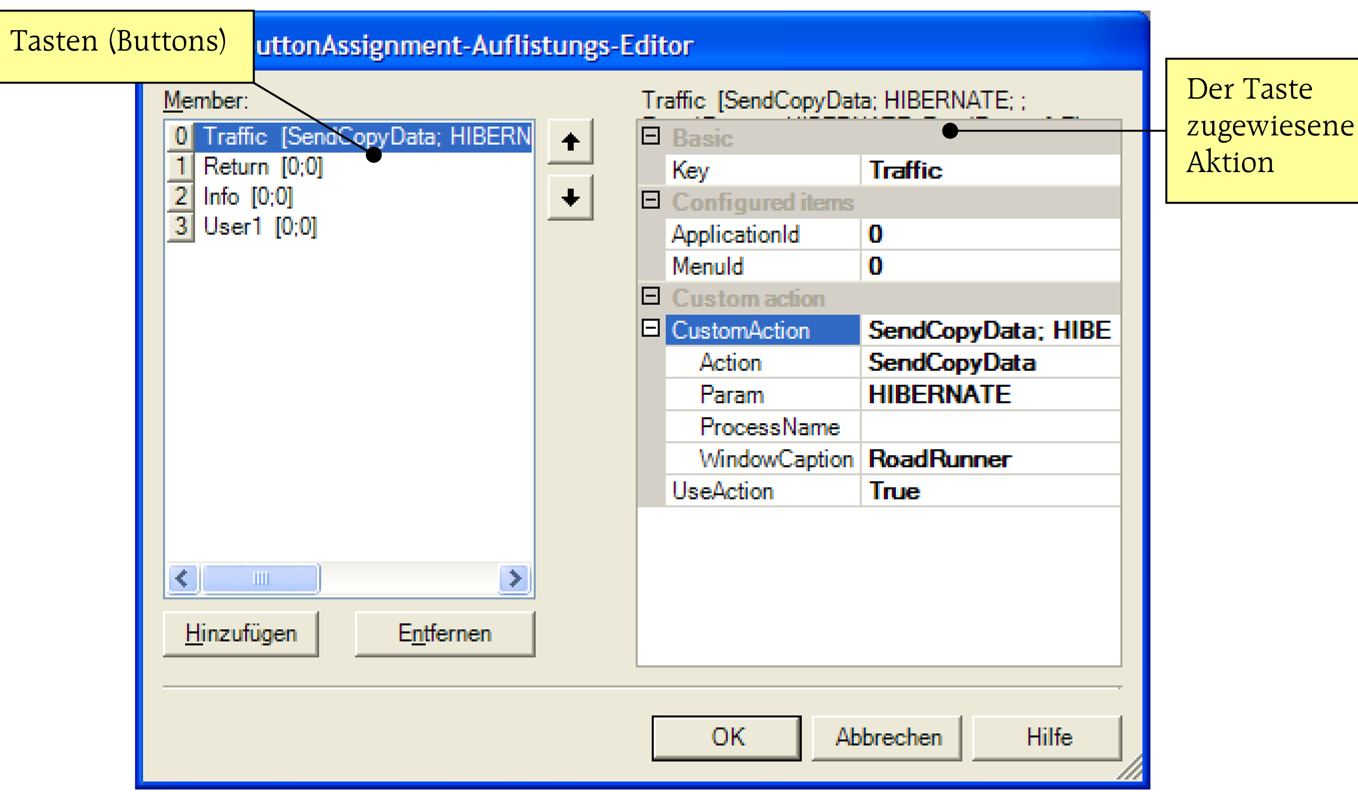


Abbildung 6: Konfiguration der Tasten des ccStick

Auf der Linken Seite des Dialogs werden die definierten Tasten gelistet und es können neue eingefügt bzw. bestehende gelöscht werden. Wird eine Taste selektiert, erscheint die zugewiesene Aktion auf der rechten Seite.
Auch hier ist eine Unterkategorisierung der besseren Übersicht wegen vorhanden.

Basic

Hier wird die entsprechende Taste angegeben.

Configured Items

Hier kann bestimmt werden, welcher Menüpunkt einer konfigurierten Applikation beim betätigen der Taste ausgeführt werden soll. Dabei müssen die Id der Applikation und des Menüpunktes eingetragen werden. Diese Informationen können frei in der Konfiguration von Applikationen eingestellt werden. Um diese Variante zu nutzen muss die Eigenschaft *UseAction* auf false gesetzt sein.

CustomAction
Hier wird definiert, ob eine andere Aktion als das Ausführen einer konfigurierten Applikation stattfindet. Dazu muss die Eigenschaft *UseAction* auf **true** gesetzt sein.

Im oben gezeigten Bild wird beim drücken der Taste „**Traffic**" die Applikation **RoadRunner** angewiesen den Rechner in der Ruhezustand (Hibernate) zu versetzen.

Details für das definieren von direkten Aktionen gibt es im Kapitel 8.

7.4 Car2PC Konfiguration

Die Anbindung des Car2PC Adapters dient nicht der Steuerung von MWI sondern der einfachen Konfiguration der Tasten des Adapters.

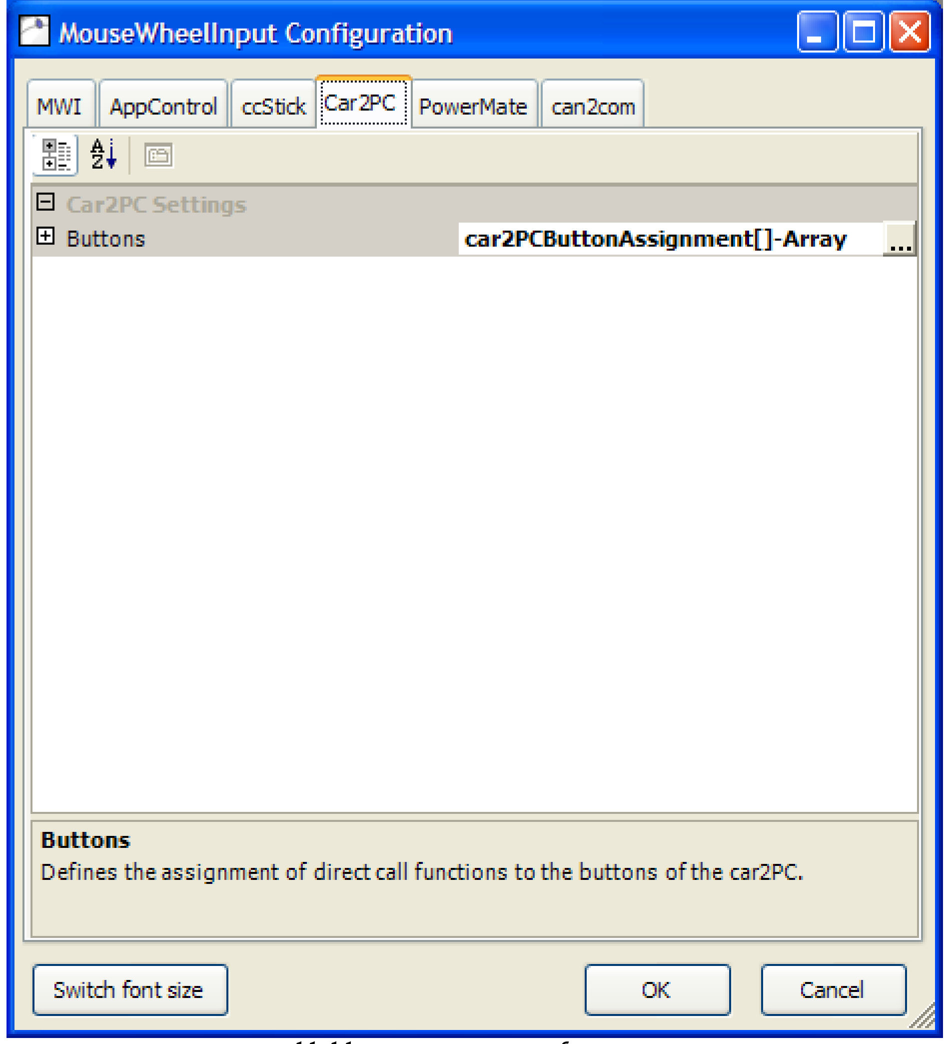


Abbildung 7: Car2PC Konfiguration

7.4.1 Unterkategorie „Car2PC Settings“

Buttons

Aus dem oben genannten Grund gibt es nur Tasten zu definierten, die eine bestimmte Aktion (wie z.B. das Abspielen des nächsten Liedes im MP3 Player) ausführt.
Die Konfiguration ist dabei die gleiche wie unter Unterkategorie „Button Assignments" beschrieben und erfolgt in dem gleichen Dialogfenster.

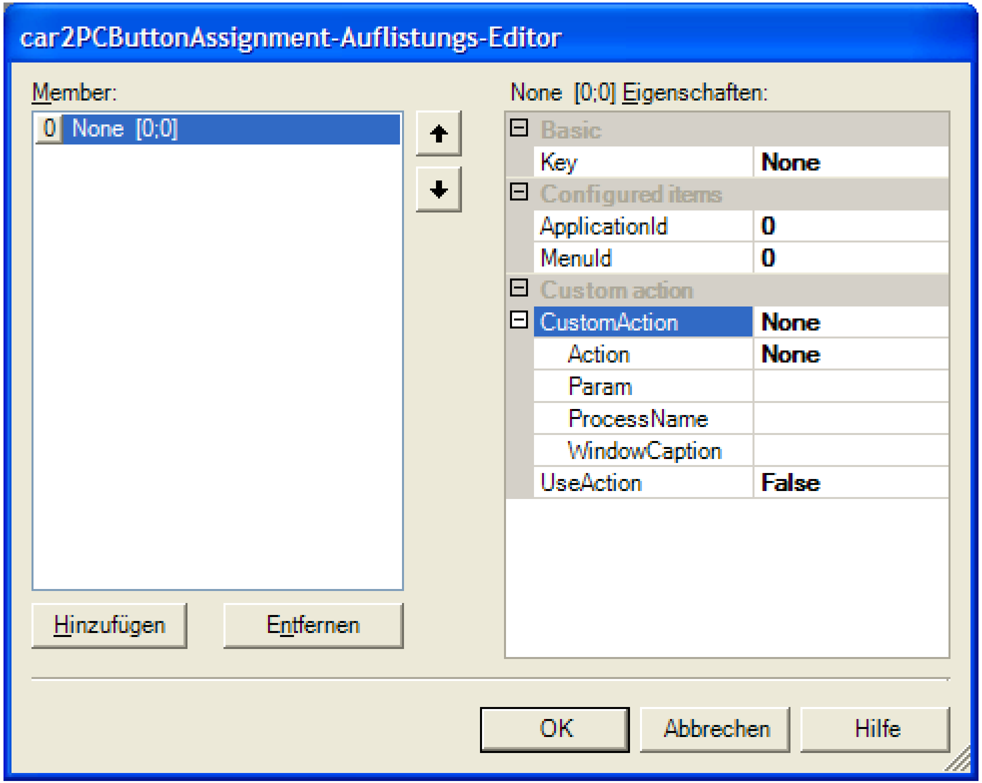
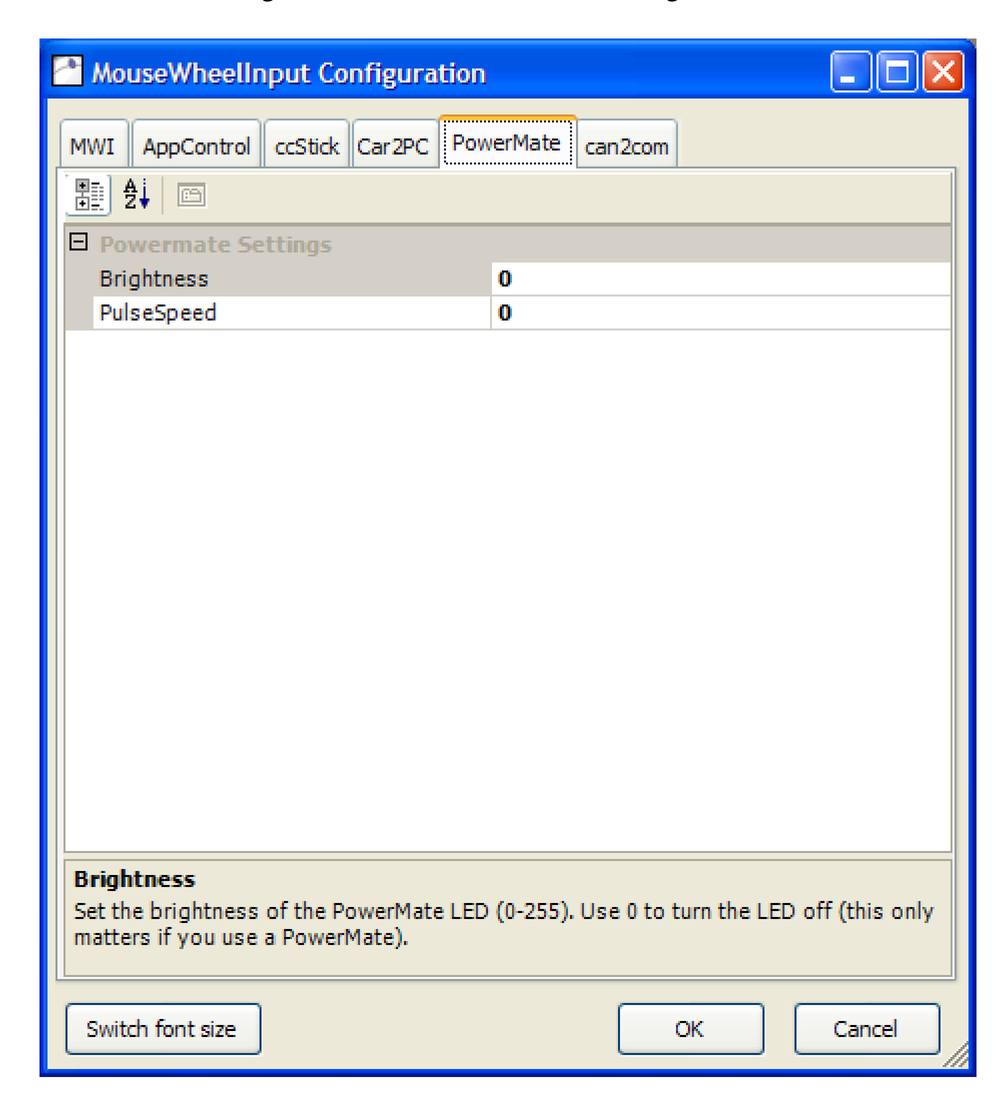


Abbildung 8: Dialog zum Konfigurieren der Tasten des Car2PC Adapters

7.5 PowerMate Konfiguration

Hier werden die Einstellungen des Griffin PowerMate vorgenommen.



7.5.1 Unterkategorie „Powermate Settings“

Da der PowerMate in der Funktion eingeschränkt ist. Gibt es hier nicht viel einzustellen.

Brightness

Beschreibt die einzustellende Helligkeit der LED des PowerMate. Wird hier eine 0 eingetragen, wird die LED abgeschaltet.

PulseSpeed

Beschreibt die Puls Geschwindigkeit der LED im PowerMate. Wird hier eine 0 eingetragen, leuchtet die LED permanent.



8 Konfiguration von Aktionen

Neben der Möglichkeit die Aktionen bestimmter Applikationen auszuführen, können in AppControl direkte Aktionen (wie das Senden von Tasten) definiert werden.
Dazu dient der Abschnitt **CustomAction** im unten gezeigten Dialog.

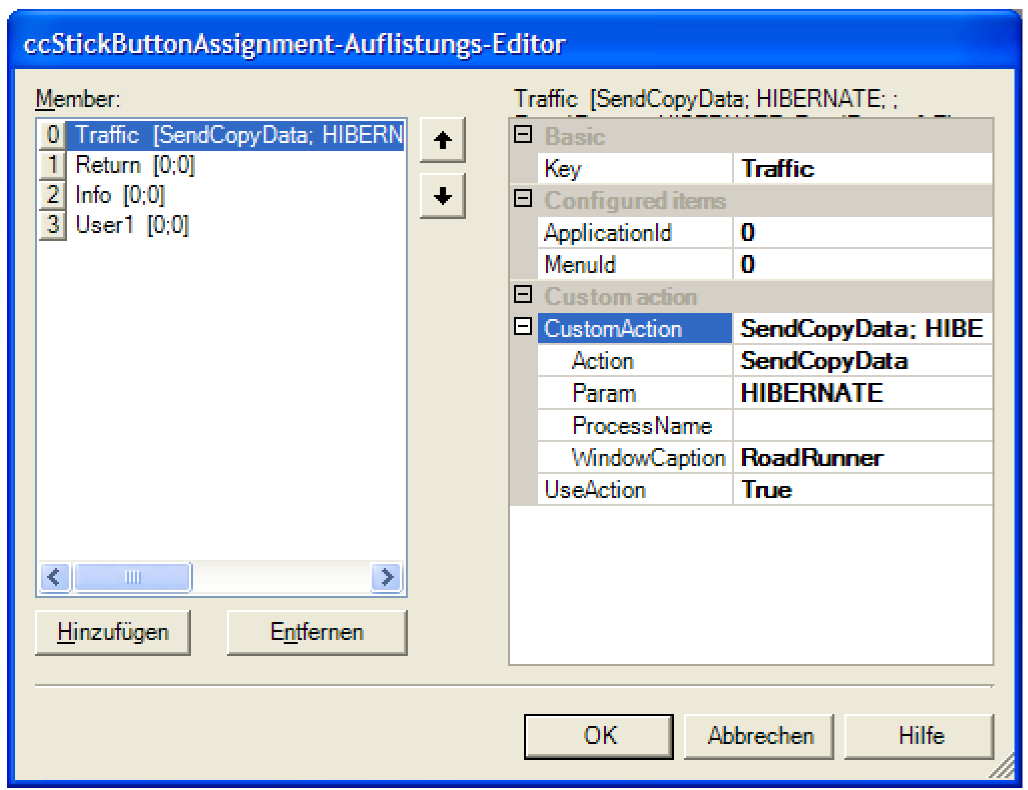


Abbildung 9: Definieren von Aktionen Dialog

Action

Hier wird definiert, welcher Mechanismus eingesetzt werden soll. Zur Auswahl stehen hierfür:

Aktion	Bedeutung
None	Keine Aktion.
SendMessage	Es wird eine Windows Nachricht (wie z.B: WM_COMMAND versendet). Dazu werden in Eigenschaft *Param* die Nachricht und ihre Parameter in folgendem Format übergeben: `Msg;wParam;lParam  z.B.: 0x786;0x0;0x33` Um Windows Nachrichten zu senden, muss man die Nachrichten und deren Parameter kennen.
PostMessage	Arbeitet so wie SendMessage.

SendCopyData	Es werden Daten mit WM_COPYDATA zur Applikation übertragen. Diese Variante ist für Programme wie z.B. RoadRunner von Bedeutung.
SendKey	Es werden Tastendrücke versendet. Die zu sendende(n) Taste(n) werden in *Param* eingetragen. Eine Liste der möglichen Tasten gibt es im Kapitel 9.3.
RunScript	Es wird ein Skript ausgeführt. Das auszuführende Skript wird mit Pfad und Dateinamen in *Param* übergeben. Diese Aktion geht nicht, wenn keine Gold Lizenz vorliegt oder *EnableScripting* in der AppControl Konfiguration auf false gesetzt wurde.

Param

Enthält immer den Parameter abhängig von der *Action*.

ProcessName

Definiert den Namen des Prozesses. Dieser sollte immer dann verwendet werden, wenn der Fenstertitel (*WindowCaption*) nicht zur Verfügung steht oder sich im Laufe der Zeit verändert. Der Name entspricht hierbei der Anzeige im Task-Manager unter Prozesse.

WindowCaption

Definiert den Titel des Fensters der Applikation, an die die Aktion gesendet wird. Hier kann ein * (Stern) eingetragen werden um die Aktion an das jeweils aktive Fenster zu senden).

Das verwendet von SendKey Aktionen ist oft am einfachsten, kann aber zu Problemen führen, wenn MWI das entsprechende Fenster nicht in den Vordergrund holen kann. Bei Applikationen wie RoadRunner oder FreeDrive sollte immer die Aktion SendCopyData verwendet werden, da dies die sicherste Art und Weise ist diese Applikationen zu steuern.

9 Konfiguration von Applikationen

Um Applikationen zu konfigurieren, die im AppControl-Fenster erscheinen sollen, reicht ein Klick auf den Menüpunkt **Configure Applications** im Kontext-Menü das hinter dem TaskBar Symbol liegt und auch rechten Mausklick erscheint.
Eine weitere Möglichkeit ist es, die Layout-Vorschau in der AppControl Konfiguration zu Doppelklicken.
Daraufhin erscheint das folgende Fenster:

Abbildung 10: Konfiguration von Applikationen

Der Baum auf der linken Seite zeigt alle konfigurierten Applikationen an und entspricht damit dem Verzeichnisbaum im Pfad der im *ConfigPath* angegeben wurde.
Der hier dargestellte Name entspricht dem Verzeichnisnamen und hat keine Bedeutung zu dem Namen der in AppControl angezeigt wird.
Die Applikationen mit ausgegraut dargestellten Symbolen sind gesperrt (*Enabled* = false) und werden in AppControl nicht dargestellt. Dies ist die bessere Alternative zum Löschen von Applikationen.

Auf der rechten Seite werden die Eigenschaften der aktuell im Baum angewählten Applikation angezeigt. Hier können die folgenden Eigenschaften eingestellt werden:

AppPath

Beschreibt den Pfad der Ausführbaren Datei, die an die Applikation zu übergebenden Parameter und das Arbeitsverzeichnis. Wird in der Eigenschaft Arguments das Schlüsselwort **$param$** angegeben, so wird der als Argument der Inhalt der jeweiligen *Param* Eigenschaft des ersten Menüeintrags übergeben.

Enabled

Definiert ob eine Applikation zur Darstellung im AppControl-Fenster freigegeben ist. Steht der Wert auf **false**, wird die Applikation nicht berücksichtigt.

Id

Definiert die (eindeutige) Id der Applikation. Diese ist frei wählbar und dient dazu diese Applikation für den direkten Aufruf über externe Tasten anzubieten. Bei der Vergabe der Id's ist darauf zu achten, dass eine Id nicht an zwei Applikationen vergeben wird.

Index

Definiert die Position an der diese Applikation im AppControl-Fenster erscheinen soll. Dieser Index kann relativ einfach verändert werden, wenn die Tastenkombination Strg-↑ bzw. Strg-↓ im Baum betätigt werden (alle nachfolgenden nicht gesperrten Applikationen werden automatisch neu nummeriert).

Name

Definiert den Namen der Applikation, der in der Titelleiste angezeigt werden soll.

Special

Definiert eine Spezial-Aktion, wenn diese Applikation selektiert wird. Derzeit stehen **Tasklist** und **MWI** zur Verfügung. **Tasklist** bietet im Menü eine Liste der laufenden Prozesse zur Auswahl an, **MWI** bringt das MWI-Fenster zur Zeicheneingabe zum Vorschein (wichtig wenn die **UseOnClick** Einstellung dies verhindert).
Die Taskliste wird in 60 Sekunden Intervallen aktualisiert.

Symbol

Definiert welches Symbol im AppControl-Fenster für diese Applikation angezeigt werden soll. Hier empfiehlt es sich ebenfalls PNG Dateien zu verwenden, da dieses Format transparente Bereiche unterstützt.

Menu

Definiert die Liste der Aktionen die auf diese Applikation durchgeführt werden kann. Hat die Applikation mehr als einen Menüeintrag, wird die Liste im Menü angezeigt; hat sie nur einen Menüeintrag definiert, wird dieser direkt ausgeführt.
Zum Bearbeiten der Liste wird ein separater Dialog angezeigt.



9.1 Bearbeiten von Menüeinträgen

Zum Bearbeiten von Menüeinträge (oder auch Aktionen einer Applikation) wird ein eigener Dialog zur Verfügung gestellt:

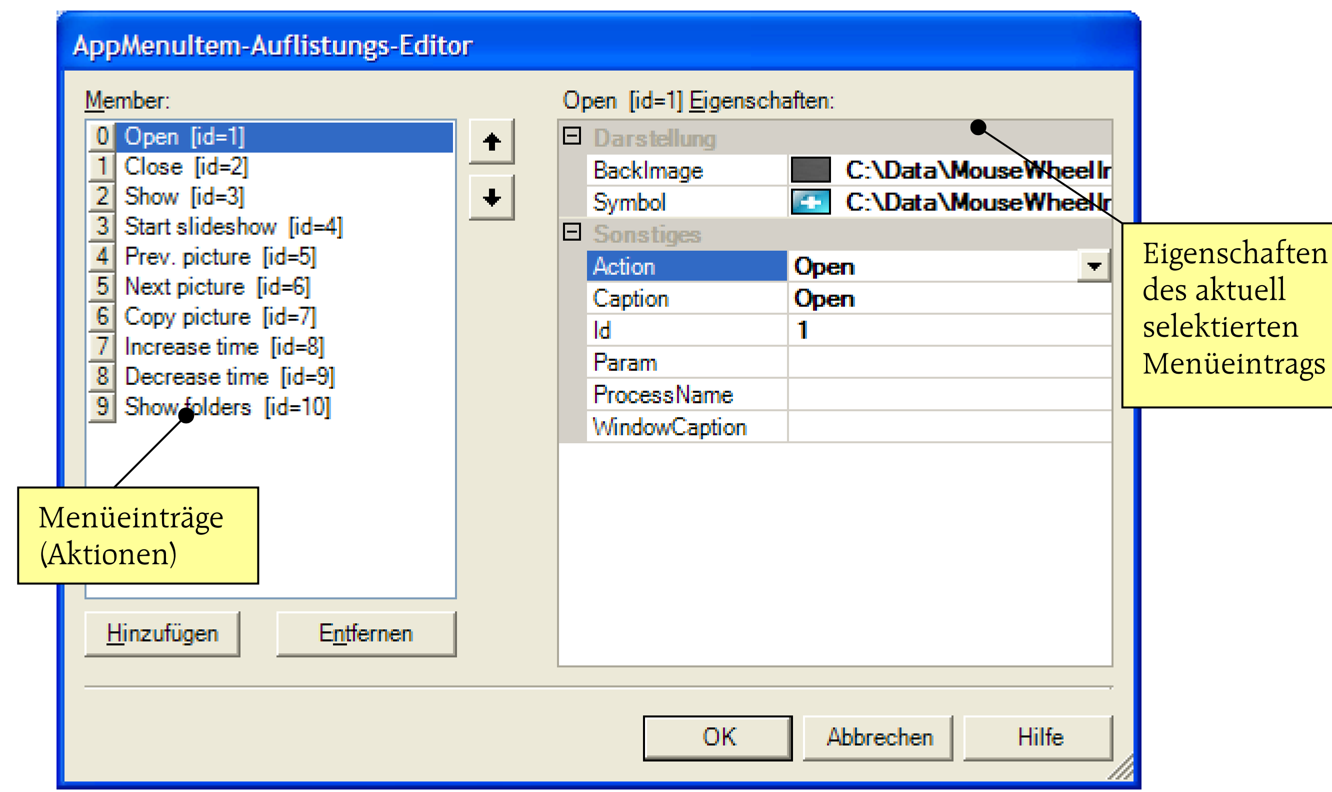


Abbildung 11: Menüeinträge zu einer Applikation

Auf der linken Seite werden die definierten Menüeinträge aufgelistet. Dort können neue hinzugefügt oder bestehende gelöscht werden.

Auf der rechten Seite sind die Eigenschaften jedes Menüeintrags ersichtlich. Folgende Optionen können eingestellt werden:

BackImage

Definiert das Hintergrundbild eines Menüeintrages. Ist dieser nicht definiert, wird ein evtl. von der AppControl-Konfiguration vorgegebener Menühintergrund verwendet.

Symbol

Gibt an welches Symbol (Icon) für diesen Menüeintrag verwendet werden soll.

Action

Definiert welche Aktion ausgeführt werden soll wenn dieser Menüeintrag ausgeführt wird. Hierfür sind dieselben Einstellungen möglich wie unter Kapitel 8 beschrieben aber auch noch die zusätzlichen:

Aktion	Bedeutung
Open	Öffnet die unter **AppPath** definierte Applikation (startet sie) ungeachtet ob diese bereits läuft oder nicht.
OpenOnce	Öffnet die unter **AppPAth** definierte Applikation (startet sie) allerdings nicht wenn diese bereits läuft. In diesem Fall wird die laufende Instanz in den Vordergrund geholt.
Close	Schließt die Instanz der unter **AppPath** definierten Applikation unter Verwendung der WM_CLOSE Windows Nachricht.
Kill	Killt die Instanz der unter **AppPath** definierten Applikation aus dem Prozessraum.
Show	Bringt die Instanz der unter **AppPath** definierten Applikation in den Vordergrund.
ConfigureMe	Ruft den Dialog zum Konfigurieren von Applikationen auf.
ConfigureMWI	Ruft den Dialog zum Konfigurieren von MWI auf.
SwitchConfig	Schaltet eine Konfiguration um (lädt die Applikations Konfigurationen aus dem in **Param** angegebenen ConfigPath). Diese Aktion benötigt keinen Prozessnamen oder Fenstertitel. Diese Funktion ist nur mit der Gold Lizenz verfügbar.
SendMouseClick	Sendet einen Mausklick. Wenn **Param** den Wert **rclick** enthält einen rechten Klick, einen linken wenn es **lclick** enthält. Diese Funktion ist global und benötigt daher keine Angabe des Prozessnamens oder Fenstertitel.

Caption

Definiert den Titel des Menüeintrags der zur Darstellung gebracht wird.

Id

Definiert die Id für diesen Menüeintrag. Diese ist frei wählbar und dient dazu diese Applikation für den direkten Aufruf über externe Tasten anzubieten. Bei der Vergabe der Id's ist darauf zu achten, dass eine Id nicht an zwei Menüeinträge vergeben wird.

Param

Enthält immer den Parameter abhängig von der *Action*.

ProcessName

Siehe *ProcessName* in Kapitel 8.

WindowCaption

Siehe *WindowCaption* in Kapitel 8.

9.2 Neue Applikationen konfigurieren

Die einfachste Art und Weise neue Applikationen zu konfigurieren ist es die Applikation aus dem Datei-Explorer in den Baum zu ziehen und dort loszulassen (Drag & Drop).
MWI erstellt daraufhin ein neues Unterverzeichnis im aktuellen Konfigurationspfad, und hinterlegt dort die Konfigurationsdatei. Das Symbol wird aus dem Icon der Applikation extrahiert (sofern vorhanden) und im PNG Format gespeichert.
Außerdem erstellt MWI drei Menüeinträge mit den Aktionen *Open*, *Show* und *Close*..

Danach kann die Applikation den Bedürfnissen angepasst werden.

Wenn eine Konfiguration auf einem anderen Rechner vorliegt, kann diese einfach durch kopieren in das Konfigurationsverzeichnis (ConfigPath) erstellt werden. Während der Laufzeit von MWI wird eine solche Aktion observiert und die kopiert Applikation steht sofort zur Verfügung.



9.3 Automatische Übersetzung der Texte

Normalerweise liegen die Applikationskonfigurationen in Englischer Sprache vor. Damit nun nicht jede konfigurierte Applikation per Hand übersetzt werden muss, kann im Ausführungsverzeichnis eine Textdatei mit dem Namen **Translation.txt** angelegt werden, die alle verwendeten Text übersetzt.
Der Inhalt der Datei kann beispielsweise folgender sein:

```
Open=Öffnen
Close=Schließen
Show=Zeigen
Bring to front=Zeigen
Play=Abspielen
Stop=Anhalten
Pause=Pause
```

Aufgrund dieser Liste werden die Texte Open, Close, Show etc. die sich innerhalb einer Applikationskonfiguration finden in das Deutsche Equivalent übersetzt. Dabei steht links vom = Zeichen der originale Text, rechts davon die zu verwendende Übersetzung.

Der Inhalt und die Länge dieser Datei sind nicht eingeschränkt.

Somit können Applikationskonfigurationen unabhängig von der Sprache weitergegeben werden.

10 Tastaturbefehle

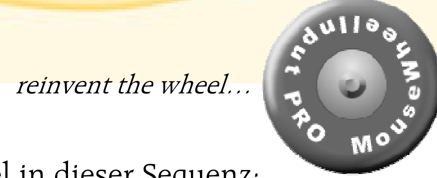
Neben den normalen alphanumerischen Zeichen können für die Aktion *SendKey* auch folgende Funktionstasten verwendet werden.

COMMA (als Ersatz für ',')
PIPE (als Ersatz für '|')
BACK
TAB
CLEAR
RETURN
SHIFT
CTRL
ALT
PAUSE
CAPITAL
ESCAPE
SPACE
PRIOR
NEXT
END
HOME
LEFT
UP
RIGHT
DOWN
SELECT
PRINT
EXECUTE
SNAPSHOT
INSERT
DELETE
HELP
STARTKEY
CONTEXTKEY
NUMPAD0 .. NUMPAD9
MULTIPLY
ADD
SEPARATOR
SUBTRACT
DECIMAL
DIVIDE
F1 .. F24
NUMLOCK
VK_BROWSER_BACK
VK_BROWSER_FORWARD
VK_BROWSER_REFRESH
VK_BROWSER_STOP
VK_BROWSER_SEARCH
VK_BROWSER_FAVORITES
VK_BROWSER_HOME
VK_VOLUME_MUTE
VK_VOLUME_DOWN
VK_VOLUME_UP
VK_MEDIA_NEXT_TRACK
VK_MEDIA_PREV_TRACK
VK_MEDIA_STOP
VK_MEDIA_PLAY_PAUSE
VK_LAUNCH_MAIL
VK_LAUNCH_MEDIA_SELECT
VK_LAUNCH_APP1
VK_LAUNCH_APP2
ICO_F17
ICO_F18
ICO_HELP
ICO_00
ICO_CLEAR
OEM_1 .. OEM_8
OEM_SCROLL
OEM_PLUS
OEM_COMMA
OEM_MINUS
OEM_PERIOD
OEM_RESET
OEM_JUMP
OEM_PA1
OEM_PA2
OEM_PA3
OEM_WSCTRL
OEM_CUSEL
OEM_ATTN
OEM_FINNISH
OEM_COPY
OEM_AUTO
OEM_ENLW
OEM_BACKTAB
OEM_CLEAR
OEM102
ATTN
CRSEL
EXSEL
EREOF
PLAY
ZOOM
NONAME
PA1

Es können auch ganze Texte gesendet werden, indem alle Zeichen dieses Textes mit Komma getrennt eingegeben werden:

```
H,a,l,l,o, ,M,W,I
```

Dies sendet den Text „*Hallo MWI*" an ein festgelegtes Fenster. Um (Funktions-) Tasten zu kombinieren, werden diese ebenfalls aneinander gereiht: Um die Kombination Strg+S zu senden wird einfach die Notation „`Ctrl,S`" verwendet.
In machen Situationen kann es notwendig sein die gedrückt / losgelassen Information zu steuern. Dazu kann MWI das „loslassen" der Tasten **Strg**, **Alt**, **Shift** mit einem



vorangestellen Minus Zeichen (-) kontrollieren, wie zum Beispiel in dieser Sequenz: „**`Alt,Tab,Tab,Tab,-Alt`**"

11 Problembehandlung

Bei Problemen die dazu führen das MWI unstabil läuft, ist stets zu prüfen, ob nicht zusätzliche Treiber im Konflikt mit MWI stehen.

11.1 Bei Verwendung vom Mausrad

Ist darauf zu achten, dass keine zusätzlichen, vom Hersteller beigefügten Maustreiber installiert sind. In manchen Fällen führ dies zum Absturz von MWI.

11.2 Bei Verwendung des PowerMate

Auch hier sollten die standard Treiber deinstalliert werden, da sich der in MWI integrierte Treiber mit dem des PowerMate nicht verträgt.
Um festzustellen ob der PowerMate richtig arbeitet, ist das Testprogramm *TestPowerMate.exe* zuständig. Sollte sich hier keine Reaktion zeigen, liegt evtl. ein Hardwarefehler vor.

11.3 Bei Verwendung des ccStick

Wenn MWI läuft darf das ccStick Testprogramm nicht gestartet werden und umgekehrt. Das führt zu einer Fehlfunktion in beiden Applikationen.



12 Unterstützte Grafik Formate

MWI unterstützt, durch das .NET Framework, die folgenden Grafik Formate:

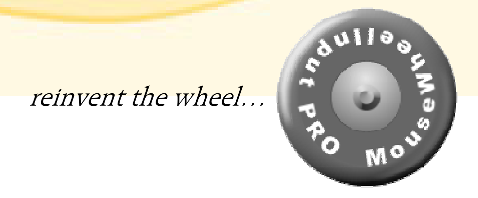
- BMP
- JPEG
- PNG
- WMF
- GIF
- EMF
- TIF

Nicht alle Formate eignen sich gleichermaßen. Die besten Ergebnisse werden mit dem PNG Format erzielt, weil dieses transparente Bereiche über den so genannten Alpha Channel realisieren kann und durch eine nicht verlustbehaftete Kompression die beste Qualität der Bilder liefert.

13 Dokumenthistorie

Folgende Änderungen wurden an diesem Dokument durchgeführt:

Version	Änderungen	Datum
1.0	Erste Version	23.05.2006



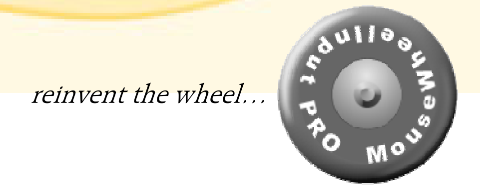
14 Bilder Verzeichnis

Abbildung 1: MWI-Fenster 5
Abbildung 2: AppControl-Fenster 5
Abbildung 3: VolumeControl-Fenster 6
Abbildung 4: MWI Konfigurations-Fenster 7
Abbildung 5: DefaultWheelAction Konfiguration 17
Abbildung 6: Konfiguration der Tasten des ccStick 21
Abbildung 7: Car2PC Konfiguration 22
Abbildung 8: Dialog zum Konfigurieren der Tasten des Car2PC Adapters 23
Abbildung 9: Definieren von Aktionen Dialog 25
Abbildung 10: Konfiguration von Applikationen 27
Abbildung 11: Menüeinträge zu einer Applikation 29

15 Verweise

Hier noch einmal die Liste der im Dokument referenzierten Hard- und Software:

- Can2com
- Car2PC
- ccStick
- Griffin PowerMate
- FreeDrive
- RoadRunner
- Flash Filme



16 Index

$

$param$ · 28

.

.NET Framework 1.1 · 4

A

Action · 17, 25, 30, 31
alphanumerischen Zeichen · 33
Alt · 33
ALT · 33
animiert · 16
AppControl · 4, 5, 14, 25, 27
AppPath · 28
Audiokanal · 13

B

BackgroundColor · 9, 15
BackgroundImage · 9, 15
BackImage · 29
Basic · 21
Baum · 27
below · 12
Betriebssystem · 4
BMP · 35
Brightness · 24
Buttons · 20, 23

C

Caption · 30
CaptionAlignment · 16
CaptionLocation · 16
CaptionSize · 16
Car2PC · 22
ccStick · 4, 19, 20
Close · 30, 31
ConfigPath · 14, 31
Configuration · 7
Configure Applications · 14, 27
Configured Items · 21
ConfigureMe · 30
ConfigureMWI · 30
Controller · 19
ControlVolume · 13
CTRL · 33
CustomAction · 22, 25

D

DefaultWheelAction · 17
Dictionary · 12
DictionaryAppendSpace · 12
DictionaryBackgroundColor · 12
DictionaryBackgroundImage · 12
DictionaryForegroundColor · 12
DictionaryOpacity · 12
DictionaryPosition · 12
DoubleclickTime · 11
Drag & Drop · 31
DynamicPosition · 9

E

Eingabemedium · 19
Einträge · 12
EMF · 35
Enabled · 27, 28
EnableScripting · 17
EnableSound · 12
Endlosband · 16
ExcludeFromTaskList · 13

F

Flash · 4
FocusColor · 15
FocusLineWidth · 15
FocusRectImage · 15
FocusRectRadius · 15
Font · 9, 15
ForegroundColor · 9, 15
FreeDrive · 26

G

GIF · 35
Gold Lizenz · 19
Gold-Edition · 14
Grafiken · 12
Griffin PowerMate · 4
Gruppierung · 8

H

HideTime · 18
Hintergrund Applikationen · 13

I

Icon · 29
Icons · 15

iD · 19
Id · 28, 30
ImageSize · 15
InBar · 17
Index · 28
InMenu · 17
InputKey · 20
Installationsverzeichnis · 15
InvertSelectedChar · 9
InvokeTime · 11
IOWarrior · 20

J

JPEG · 35

K

Kill · 30
Konfigurationen · 14
Konfigurationsfenster · 7
kurzen Klick · 5

L

langen Klick · 5
Lautstärke · 6
Lautstärke Regelung · 13
Layout · 8
Layout-Vorschau · 14
Letters · 9
Location · 9, 16

M

Maustreiber · 34
MaxDictionaryList · 12
Menu · 28
MenuBackImage · 16
Menüeinträge · 29
MenuKey · 20
Millisekunden · 18
Mini · 19
Minus Zeichen · 34
Mode · 19
MouseKey · 20
MouseWheelInput · 4
MP3 Player · 23
MWI · 5, 28
MWIProSetup.zip · 4
MyConfigs · 15

N

Name · 28

O

OnClick · 17
OnTurnLeft · 17
OnTurnRight · 17
Opacity · 10, 16
Open · 30, 31
OpenOnce · 30

P

Param · 17, 26, 31
Pfeile · 16
Pin 1 · 20
PNG · 28, 31, 35
Position · 16
PostMessage · 25
PowerMate · 24, 34
ProcessName · 17, 26, 31
Prozesse · 13
PulseSpeed · 24

R

rechteckig · 15
RevokeTime · 11
right · 12
RoadRunner · 22, 26
Ruhezeit · 11
RunScript · 26

S

SelectionLineWith · 10
SelectionRectRadius · 10
SendCopyData · 26
SendKey · 26, 33
SendMessage · 25
SendMouseClick · 30
Setup.msi · 4
Shift · 33
SHIFT · 33
Show · 30, 31
ShowArrows · 16
Size · 10, 16
skaliert · 15
SlowDown · 11
Soundeffekte · 12
Special · 28
Sprachausgabe · 12
Standard · 19
Stimme · 12
Strg · 33
SwapWheelDirection · 17
SwitchConfig · 30
SwitchToDictionaryTime · 11
Symbol · 28, 29
Symbole · 15



T

tabellarische Darstellung · 8
TaskBar · 27
Tasklist · 28
Taskliste · 13
Task-Manager · 13
Tastaturkommandos · 14
Tasten · 21, 30
Tasten-Aktionen · 21
TestPowerMate.exe · 34
Testprogramm · 34
TIF · 35
Titelleiste · 16
Traffic · 22
Translation.txt · 32

U

UseAction · 21, 22
UseAnimation · 10, 16
UseAppControl · 10
UseCan2Com · 10
UseCar2PC · 10
UseCcStick · 10
UseFading · 17
UseHorizontalMenu · 17
UseMouseHook · 11
UseOnClick · 11
UsePowerMate · 11
UseVoice · 12

V

Verzeichnisbaum · 27
VoiceTime · 11
VolumeColor · 13
VolumeControl · 6, 20
VolumeKey · 20
VolumeOpacity · 13

W

WindowCaption · 17, 26, 31
WMF · 35
Wörterbuch · 5, 12

Z

Zeitspanne · 18